

Express Mailing Label No.: ER211528425US

PATENT APPLICATION

IBM Docket No.: TUC920030086US1

Kunzler & Associates Docket No.: 1500.2.21

UNITED STATES PATENT APPLICATION

of

RALPH T. BEESTON,

KIRBY G. DAHMAN,

GAVIN S. JOHNSON

and

TRANG T. LE

for

**APPARATUS, SYSTEM AND METHOD TO ESTIMATE MEMORY
FOR RECOVERING DATA**

APPARATUS, SYSTEM, AND MEHTOD TO ESTIMATE MEMORY FOR RECOVERING DATA

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] This invention relates to estimating the size of a data block in a data buffer, and more particularly, relates to creating an estimate for allocating memory to recover the data block from the data buffer.

DESCRIPTION OF THE RELATED ART

[0002] Computer systems regularly write data blocks from their internal memory to a data buffer of an external destination node. Conventional destination nodes may include storage devices, network connections, and transmission devices. The data buffer stores the data block until the data block is written to the destination medium and the transaction is completed. Conventional destination mediums may include magnetic tape, a hard disk drive, an optical storage drive, or a router. The computer system considers the data block write transaction as complete when the data block is written to the data buffer. The memory locations where the data block was stored in the computer system, such as RAM, may be deallocated and overwritten.

[0003] If an error occurs that prevents the data block from being written to the destination medium after the location where the data block was stored in computer memory space has been deallocated, the data block will be lost unless the computer system recovers the data block from the data buffer. The computer system recovers the data block by allocating a memory space sufficient to hold the data block in the computer's memory, such as RAM memory. The computer then copies the data block from the data buffer into the memory space.

[0004] The computer system does not retain a record of the size of the data block after the data block has been written to the data buffer. Yet the computer system must

reserve a memory space sufficiently large to contain the data block, or a portion of the data block may be lost. Computer systems frequently reserve a memory space as large as the entire data buffer to ensure that the memory space is sufficiently large to recover the whole data block.

[0005] Reserving a memory space as large as the data buffer, however, may consume significantly more memory than is required to actually recover the data block. Reserving excessive memory increases the processing time required to complete recovery of the data block, potentially putting the successful completion of the recovery at risk. Excessive reservation of memory for recovery operations may also slow down other computer functions that require memory. The degradation of computer performance due to excessive memory reservation may be exacerbated when multiple memory spaces must be allocated to recover multiple data blocks from the data buffer. Recovering multiple data blocks can even consume enough memory to completely stall computer operation.

[0006] Figure 1 is a block diagram illustrating a conventional data processing system 100. The system 100 includes a computer 105, a destination node 110, a memory module 115, a processor module 120, a data buffer 125, and a destination medium 130. The system 100 transfers data blocks from the computer 105 to the destination node 110. The transfer is typically conducted using a bus or computer network 140.

[0007] The computer 105 transfers a data block to a destination node 110, copying the data block from the memory module 115 to the data buffer 125 of the destination node 110. The destination node 110 copies the data block from the data buffer 125 to the destination medium 130 to complete the transfer.

[0008] Figure 2 is block diagram illustrating a memory map 200 of the current technology. The memory map 200 includes a memory module 115, a data buffer 125, one or more memory data blocks 205, one or more buffer data blocks 210, and an allocated memory space 215. The memory map 200 shows the disposition of data blocks in the memory module 115 and the data buffer 125 during a data recovery process.

[0009] The computer 105 transfers a memory data block 205c to the destination node 110 by copying the memory data block 205c to the buffer data block 210a in the data buffer 125 as indicated by an arrow 220. Buffer data blocks 210 typically occupy less than the whole data buffer 125. The destination node 110 writes the buffer data block 210a to the destination medium 130 to complete the transfer. If an error condition occurs before the buffer data block 210a can be completely written to the destination medium 130, the computer 105 initiates a data block recovery process to recover the buffer data block 210a.

[0010] The computer 105 reserves the allocated memory space 215 to recover the buffer data block 210a. The allocated memory space 215 must be as large as the buffer data block 210a or a portion of the buffer data block 210a will be lost. The size of the allocated memory space 215 that is typically reserved is equal to the size of the data buffer 125 to assure that no data in the buffer data block 210a is lost. The computer 105 copies the buffer data block 210a to the allocated memory space 215 as indicated by an arrow 225 to complete the data block recovery process.

[0011] As can be seen from Figure 2, the allocated memory space 215 is frequently much larger than is needed to recover the buffer data block 210a. Reserving more memory than is needed is costly. The unused portion of the allocated memory space 215 cannot be used for other computer 105 processes. Reserving a large memory space also takes more processing time than reserving a smaller space. Reserving memory for recovering buffer data blocks 210 will occasionally consume more memory space than is available in the memory module 115, causing the computer 105 to stall.

[0012] The reserving of memory space 215 for recovering buffer data blocks 210 and the memory related performance of the computer 105 may be improved if the size of the allocated memory space 215 more closely approximates the size of the recovered buffer data block 210. The computer 105 can more efficiently reserve a smaller memory space equivalent to the size of the buffer data block 210 instead of a larger allocated memory space 215 equal to the size of the data buffer. Reserving memory space equal to the buffer data

block 210 size would also reduce the total memory required by the data block recovery procedure, freeing memory from other tasks.

[0013] Accordingly a need exists for what is needed is a process, apparatus, and system that estimate the size of a data block to be recovered during an error condition. Beneficially, such a process, apparatus, and system would accelerate the allocation of memory for recovering data blocks and reduce the consumption of memory by the data block recovery procedure.

KUNZLER & ASSOCIATES
ATTORNEYS AT LAW
10 WEST 100 SOUTH, SUITE 450
SALT LAKE CITY, UTAH 84101

BRIEF SUMMARY OF THE INVENTION

[0014] The present invention has been developed in response to the present state of the art, and in particular, in response to the problems and needs in the art that have not yet been fully solved by currently available data block recovery processes. Accordingly, the present invention has been developed to provide a process, apparatus, and system for estimating the size of a data block that overcome many or all of the above-discussed shortcomings in the art.

[0015] The apparatus for estimating data block size is provided with a block estimator containing a plurality of modules configured to functionally execute the necessary steps of estimating the data block size for a data block that is to be recovered from a data buffer. These modules in the described embodiments include a communications module, a register module, and a logic module.

[0016] The communications module communicates with the data buffer and a computer. The logic module queries the data buffer and the computer through the communications module. In one embodiment, the logic module queries the size of a buffer data block in the data buffer. In an alternate embodiment, the logic module queries the size of a memory data block in the computer.

[0017] The register module stores a data block size record. The logic module sets the value of the data block size record to the greater of the current data block size record and the size of the queried data block. The logic module further generates a data block size estimate proportional to the data block size record. In one embodiment, the data block size estimate is equal to the data block size record. In an alternate embodiment, the data block size estimate is greater than the data block size record. The block estimator may communicate the data block size estimate to a computer.

[0018] A system of the present invention is also presented for estimating data block size for allocating memory to recover a data block. The system may be embodied in a data processing system. In particular, the system, in one embodiment, includes a data buffer for

storing a data block, a memory module for storing one or more data blocks, and a block estimator module that estimates the size of the data block.

[0019] The system copies a memory data block to the data buffer as a buffer data block. In one embodiment, the buffer data block is stored in the data buffer prior to being written to the destination medium. The destination medium may be a storage device such as a tape drive. The system deallocates the memory space of the memory data block after copying the data block to the data buffer. The buffer data block may also be deleted from the data buffer after the data block is written to destination node.

[0020] If an error condition occurs that prevents the buffer data block from being written to the destination node, the system may attempt to recover the data block to the memory module. To recover a data block from the data buffer, the block estimator module estimates the size of the buffer data block. The system reserves an allocated memory space in the memory module proportional to the data block size estimate. The data block is copied to the allocated memory space to complete the data block recovery.

[0021] In one embodiment, the block estimator module estimates the data block size estimate as proportional to the largest buffer data block in the data buffer. In an alternate embodiment, the data block size estimate is proportional to the largest memory data block in the memory module. In a certain embodiment, the data block size estimate is proportional to the size of a memory data block queued to be written to the data buffer. The data block size estimate may be smaller, equal to, or larger than the data block size used to create the data block size estimate.

[0022] A process of the present invention is also presented for estimating the size of a data block for recovering the data block. The process in the disclosed embodiments substantially includes the steps necessary to carry out the functions presented above with respect to the operation of the described apparatus and system. In one embodiment, the process includes querying the size of a data block and creating a data block size estimate proportional to the size of the queried data block.

[0023] In one embodiment, the queried data block is the largest data block in a data buffer. The largest block may be the largest block written since the data buffer was emptied or the largest data block in the data buffer. In an alternate embodiment, the queried data block is a data block in a memory module of a computer. In a certain embodiment, the data block size estimate is equal to the size of a data block queued to be written to the data buffer. In one embodiment, the process creates a data block size estimate equal to the size of the queried data block. In an alternate embodiment, the process creates a data block size estimate greater than the size of the queried data block. In one embodiment, the process records the size of the largest data block in the data buffer.

[0024] The present invention estimates the size of a data block in a data buffer. The data block size estimate may be used to reserve memory space for recovering the buffer data block from the data buffer. The data block size estimate is typically smaller than an estimate based on the size of the data buffer, reducing the processing time and memory required to reserve a sufficiently large memory space to recover the buffer data block. These features and advantages of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] In order that the advantages of the invention will be readily understood, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments that are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings, in which:

[0026] Figure 1 is a block diagram illustrating a data processing system of the current technology;

[0027] Figure 2 is block diagram illustrating a memory map of the current technology;

[0028] Figure 3 is a block diagram illustrating one embodiment of a data processing system of the present invention;

[0029] Figure 4 is a block diagram illustrating one embodiment of a block estimator of the present invention;

[0030] Figure 5 is a block diagram illustrating a one embodiment of destination node of the present invention;

[0031] Figure 6 is a block diagram illustrating one embodiment of a memory map of the present invention;

[0032] Figure 7 is a flow chart diagram illustrating one embodiment of a data block recovery process of the present invention;

[0033] Figure 8 is a flow chart diagram illustrating one embodiment of a data block size record generation process of the present invention; and

[0034] Figure 9 is a flow chart diagram illustrating one embodiment of a data block estimation process of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0035] Many of the functional units described in this specification have been labeled as modules, in order to more particularly emphasize their implementation independence. For example, a module may be implemented as a hardware circuit comprising custom VLSI circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices or the like.

[0036] Modules may also be implemented in software for execution by various types of processors. An identified module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions, which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module.

[0037] Indeed, a module of executable code could be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, merely as electronic signals on a system or network.

[0038] Figure 3 is a block diagram illustrating one embodiment of a data processing system 300 of the present invention. The system 300 includes a computer 105, a memory module 115, a processor module 120, a block estimator module 305, a destination node 110, a data buffer 125, a destination medium 130 and a bus or computer network 140. The data

processing system 300 illustrates the recovery of a data block transferred to the destination node 110.

[0039] If an error occurs before the buffer data block 210a is written to the destination medium 130, the computer 105 recovers the buffer data block 210a. To recover the buffer data block 210a, the block estimator module 305, in one embodiment, estimates the size of the buffer data block 210a. The block estimator module 305 reserves an allocated memory space in memory module 115 proportional to the data block size estimate. The computer 105 copies the buffer data block 210a to the allocated memory space to complete the data block recovery process.

[0040] In one embodiment, the block estimator module 305 determines the size of the largest buffer data block 210b in the data buffer 125 and creates the data block size estimate proportional to the largest buffer data block size. In an alternate embodiment, the block estimator module 305 creates the data block size estimate proportional to the size of the largest memory data block 205a in the memory module 115. In a certain embodiment, the block estimator module 305 creates the block estimator module 305 proportional to the size of the memory data block 205 created by the same application software that created the buffer data block 210a. In another embodiment, the data block size estimate is equal to the size of the largest data block. In yet another embodiment, the data block size estimate is larger than the size of the largest data block.

[0041] The data processing system 300 recovers a buffer data block 210 from the data buffer 125 of the destination node 110 if an error prevents the completion of a data write transaction to the destination medium 130. The block estimator 305 creates a data block size estimate that is typically smaller than the size of the data buffer 125. The system 300 reserves the allocated memory space to recover the buffer data block 210a proportional to smaller data block size estimate, reducing the memory space requirements of a data block recovery process.

[0042] Figure 4 is a block diagram illustrating one embodiment of a block estimator module 305 of the present invention. The block estimator module 305 includes a logic module 405, a register module 410, and a communications module 415. As described above, the block estimator module 305 creates a data block size estimate of the size of a buffer data block 210 in the data buffer 125.

[0043] The communications module 415 communicates with the computer 105 and the data buffer 125 of the destination node 110. The logic module 405 queries the computer 105 and the data buffer 125 through the communications module 415. In one embodiment, the logic module 405 queries the data buffer 125 to determine the size of the largest buffer data block 210 in the data buffer 125. The logic module 405 may create a data block size estimate proportional to the size of the largest buffer data block 210.

[0044] In an alternate embodiment, the logic module 405 queries the computer 105 to determine the size of a memory data block 205. The logic module 405 may query the size of a memory data block 205 created by the same software application that wrote the buffer data block 210a to be recovered. In a certain embodiment, the logic module 405 may query the size of the largest memory data block 205 in the memory module 115 of the computer 105.

[0045] The register module 415 saves a data block size record. In one embodiment, the data block size record represents the largest data block size that the block estimator 305 has received from querying the computer 105 and the data buffer 125. The logic module 405 may compare the size of a queried data block to the value of the register module 415. If the size of the queried data block is larger, the logic module 405 may write the size of the data block to the register module 415. In an alternate embodiment, the block estimator 305 may set the data block size record to zero prior to creating a data block size estimate.

[0046] The block estimator 305 creates a data block size estimate of the size of the buffer data block 210. To create the data block size estimate, the block estimator 305 determines which is the largest of the queried buffer data blocks 210 or memory data blocks 205 and calculates an estimate proportional to the largest data block size. In one

embodiment, the data block size estimate is equal to the size of the queried data block. In an alternate embodiment, the data block size estimate is greater than the size of the queried data block by a selected amount.

[0047] Figure 5 is a block diagram illustrating a destination node 500 of the present invention. The destination node 500 includes a data buffer 125, a destination medium 130, and a block estimator 305. The destination node 500 creates a data block size estimate to report to a computer 105 that may recover a buffer data block 210 from the data buffer 125 of the destination node 500.

[0048] The destination node 500 receives a memory data block 205c into the data buffer 125. In one embodiment, the block estimator 305 creates and maintains a data block size estimate of the largest buffer data block 210b potentially residing in the data buffer 125. In an alternate embodiment, the block estimator 305 creates a data block size estimate in response to a request for a data block size estimate or in response to an error condition. The destination node 500 may respond with the data block size estimate to a request from the computer 105 attempting to recover the buffer data block 210a. The destination node 500 may include the data block size estimate when notifying the computer 105 of an error condition.

[0049] Figure 6 is a block diagram illustrating one embodiment of a memory map 600 of the present invention. The memory map 600 includes a memory module 115, a data buffer 125, one or more memory data blocks 205, one or more buffer data blocks 210, and one or more allocated memory spaces 605. The memory map 600 illustrates the reservation of allocated memory space 605 proportional to the data block size estimate.

[0050] If an error condition occurs that prevents the data buffer 125 from copying the buffer data block 210a to the destination medium 130, the computer 105 must recover the buffer data block 210a or lose the buffer data block's 210a data. To recover the buffer data block 210a, the block estimator 305 estimates the size of the buffer data block 210a in the data buffer 125. The computer 105 reserves the allocated memory space 605a proportional

to the data block size estimate to receive the buffer data block 210a. In one embodiment, the allocated memory space 605a is at least as large as the data block size estimate. In an alternate embodiment, the allocated memory space 605a is larger by a selected amount than the data block size estimate. The computer 105 recovers the buffer data block 210a to the allocated memory space 605a.

[0051] Reserving the allocated memory space 605a equivalent in size to the buffer data block 210a typically reduces the size of the allocated memory space 605a. The smaller allocated memory space 605a may allow the reservation of additional allocated memory space 605b to recover a second buffer data block 210b, reducing the risk that the computer 105 will stall or the data block recovery process will fail because of insufficient memory. The smaller allocated memory space 605a may also reduce the processing time required to reserve the allocated memory space 605a.

[0052] Figure 7 is a flow chart diagram illustrating one embodiment of a data block recovery process 700 of the present invention. The data block recovery process 700 estimates the size of the buffer data block 210a to be recovered during a data block recovery process 700 and reserves the allocated memory space 605a proportional to the data block size estimate. Although for purposes of clarity the process 700 is depicted in a certain sequential order, execution may be conducted in parallel and not necessarily in the depicted order.

[0053] The computer 105 writes 705 the memory data block 205c to the buffer data block 210a of the data buffer 125. The process 700 determines 710 if an error condition occurred that prevented the destination node 110 from copying the buffer data block 210a to the destination medium 130. If no error condition occurred, the data block recovery process 700 ends 730.

[0054] If an error condition occurs indicating that the data buffer failed to write the buffer data block 210a to the destination medium 130, the data block recovery process 700 proceeds to estimate 715 the size of the buffer data block 210a. The data block recovery process 700 reserves 720 the allocated memory space 615 proportional to the data block size

estimate, reducing the size of the memory space required for the data block recovery process 700. The computer 105 recovers 725 the buffer data block 210a to the allocated memory space 605a. The risk of the computer 105 stalling because excessive allocated memory space 605 is reserved for the data block recovery process is reduced. The processing time needed to reserve allocated memory space 605 may also be reduced, although the processing time required to generate a data block size estimate may be increased.

[0055] Figure 8 is a flow chart diagram illustrating one embodiment of a data block size record generation process 800 of the present invention. The data block size record generation process 800 generates a data block size record of the largest buffer data block 210 in the data buffer 125. The data block size record may be used for estimating 715 the data block size estimate during the data block recovery process 700. Although for purposes of clarity the process 800 is depicted in a certain sequential order, execution may be conducted in parallel and not necessarily in the depicted order.

[0056] The data buffer 125 receives 802 a memory data block 205. The block estimator 305 determines 805 if all previously received buffer data blocks 210 have been written from the data buffer 125. If the previous buffer data blocks 210 have not been written, the process 800 loops to determine 820 if the received memory data block 205 size is greater than the data block size record. If the previously received buffer data blocks 210 have been written, the block estimator 305 proceeds to set 810 the data block size record to the size of the received data block.

[0057] The block estimator 305 determines 820 if the memory data block 205 written to the data buffer 125 is larger than the data block size record. If the memory data block 205 is not larger than the data block size record, the process 800 loops to receive 802 a memory data block 205. If the memory data block 205 is larger than the data block size record, the process 800 proceeds to set 825 the data block size record to the size of the memory data block 205. The data block size record generation process 800 loops to receive 802 the memory data blocks 205.

[0058] The process 800 maintains a data block size record equal to the size of the largest possible buffer data block 210 in the data buffer 125. The data block size record provides a convenient data block size for creating the data block size estimate for the data buffer recovery process 700.

[0059] Figure 9 is a flow chart diagram illustrating one embodiment of a data block estimation process 715 of Figure 7 of the present invention. The data block size estimation process 715 creates a data block size estimate. Although for purposes of clarity the process 715 is depicted in a certain sequential order, execution may be conducted in parallel and not necessarily in the depicted order.

[0060] In one embodiment, the block estimator 305 queries 905 a buffer data block 210 in the data buffer 125. In an alternate embodiment, the block estimator 305 queries 905 a memory data block 205 in the memory module 115. The memory data block 205 may have been created by the same application program that created the buffer data block 210a. In a certain embodiment, the block estimator 305 queries 905 a memory data block 205 that is scheduled to be written to the destination node 110 that reported the error condition triggering the data block recovery process 700. In an alternate embodiment, the queried memory data block 205 is scheduled to be written to any destination node 110.

[0061] In one embodiment, the data block estimator process 715 acquires 910 the size of the queried memory data block 205. The data block estimation process 715 may select the first queried data block as the basis for a data block size estimate. Alternately, the data block estimation process 715 may select the largest queried data block 205 as basis for the data block size estimate.

[0062] The data block estimation process 715 creates 915 a data block size estimate proportional to the size of a queried data block. In one embodiment, the data block size estimate is equivalent to the size of the queried data block. In an alternate embodiment, the data block size estimate is larger than the size of the queried data block.

[0063] The data block estimation process 715 creates a data block size estimate proportional to the size of queried data block. The data block size estimate may be used by the data block recovery process 700 to recover a buffer data block 210 from a data buffer 125.

[0064] The present invention estimates the size of a buffer data block 210 in a data buffer 125. The data block size estimate may be used to reserve allocated memory space 605 for recovering the buffer data block 210 from the data buffer 125. The data block size estimate is typically smaller than an estimate based on the size of the data buffer 125, reducing the processing time and memory required to reserve a sufficiently large memory space to recover the data block.

[0065] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

[0066] What is claimed is: